Last time:

Prop I.2: Given a point A and a length BC. How do we move the length BC to the point A (even though our compass is collapsing)?

Tactic	**Reason**
1. Given points A, B and C. Want a point H so that dist A H = dist B C	Given
1'. If A = B, then BC is already at A, and we're done. Assume A ≠ B	Case split
2. Construct an equilateral triangle on AB (say), call it ABD.	Prop I.1
2'. If B = C, then dist A A = dist B C, so use A. Else B ≠ C	Case split
3. Draw a circle β with center B and passing through C	Postulate 3 and B ≠ C
4. Draw a line L through B and D	Postulate 1
4'. By the way, B ≠ D because A ≠ B, and dist AB = dist BD	From Step 2
5. Draw a line M through D and A	Postulate 1
5'. B is interior of circle β	in_Circ_of_center B β
6. Line DB and circle β intersect	Line_Circ_inter_iff, Step5' Step4
7. There are two distinct points E and F on circle β and line L	Pts_of_Line_Circ_inter Step 6
8. Draw a circle δ centered at D passing through E (or is it F???)	Postulate 3
9. Circle δ intersects the line M through A and D	Line_Circ_inter_iff (D is the center of Circle δ, and hence inside)
10. Obtain points H and I where M and δ intersect	Pts_of_Line_Circ_inter
11. use H. then argue (using chain of transitivity and symmetry of the equal sign, together with definition of a circle) that dist A H = dist B C as desired	As described.

Things we still need: A predicate for when lines intersect circles

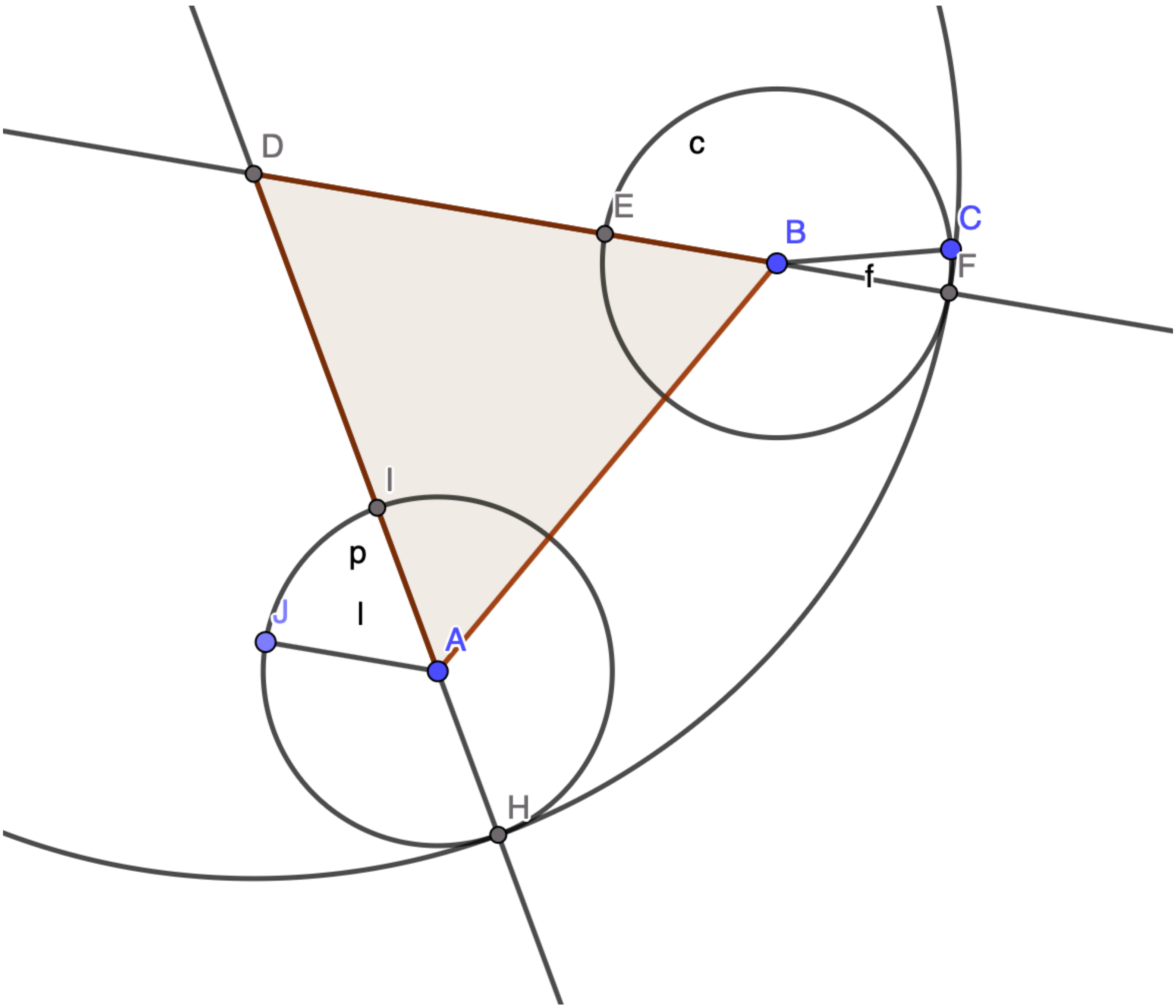
`Line_Circ_inter : Line → Circ → Prop`

Need a condition under which lines and circles do intersect, and then we need an axiom that says that if they do intersect, then they intersect at two distinct points.

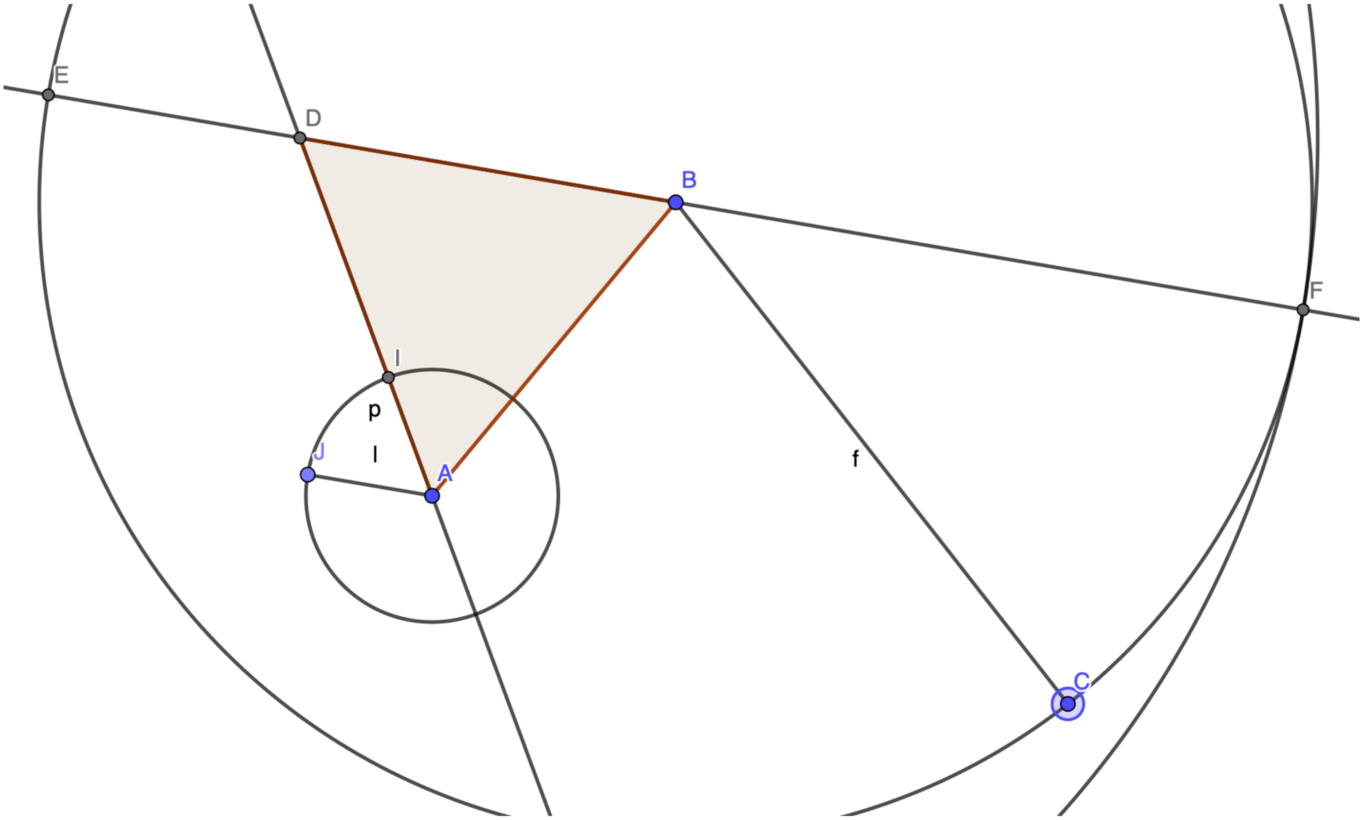
```
Line_Circ_inter_iff : ∀ (L : Line) (α : Circ), Line_Circ_inter L α ↔ ∃ (a : Pt),
    Pt_in_Circ a α ∧ Pt_on_Line a L
```

If a Line and Circle do intersect, then there are two distinct points of intersection

```
Pts_of_Line_Circ_inter :  ∀ (L : Line) (α : Circ), Line_Circ_inter L α →
    ∃ (a b : Pt), Pt_on_Line a L ∧ Pt_on_Line b L ∧ Pt_on_Circ a α ∧
        Pt_on_Circ b α ∧ a ≠ b
```



But wait a second, if I pull C a little farther from B, then the configuration flips, and `dist A H` is no longer equal to `dist B C`! What's going on??!?



Homework: Think deeply about what is going on and how we can resolve things.
Big hint: Add a new predicate of betweenness

```
InOrder : Pt → Pt → Pt → Prop
```

Where `InOrder a b c` is meant to represent the fact that all three points lie on the same line, and they occur on that line in that order, `a` then `b` then `c`.